



# Energy efficient broadcast protocols for asynchronous duty-cycled wireless sensor networks

Parmod Singh<sup>1</sup> · Yaw-Chung Chen<sup>1</sup>

Published online: 21 November 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

In asynchronous duty-cycled wireless sensor networks (WSNs), each node has its own sleep schedule that is independent of the schedules of neighboring nodes. Since each node follows its own independent sleep schedule, the probability that a node's neighbors will wake up concurrently is quite low. For this reason, each single hop broadcast must be completed through multiple unicast transmissions, and the sender node must remain awake until all of its neighboring nodes have received the broadcast packet. In this paper, we present two energy efficient broadcast protocols, namely Broadcast Progress-based Efficient Multihop Broadcast (BPEMB) protocol and Tree-based Efficient Multihop Broadcast (TEMB) protocol, for asynchronous duty-cycled WSNs. In both of the proposed protocols, the decision of whether to transmit the broadcast packet is based on the link quality information and without any forwarder's guidance list. The redundant transmissions and collisions are avoided effectively with cooperation among the neighbors, so that the node with the best link quality to the receiver node takes responsibility to forward the packet to the receiver node. With this approach, nodes that are not involved in transmitting the broadcast packet can go to sleep and save their energy. In the BPEMB protocol, the forwarding decision of transmitting the broadcast packet is also influenced by the broadcast progress information, which is piggybacked in the broadcast packet. Through extensive simulations, we demonstrate that the proposed protocols can significantly improve the broadcast efficiency through reducing the energy consumption and minimizing the broadcast latency. The evaluation results also show that both the BPEMB and TEMB protocols can substantially reduce the collisions and redundant transmissions in asynchronous duty-cycled WSNs.

**Keywords** Wireless sensor networks · Asynchronous sleep scheduling · Multihop broadcast · Energy conservation · Piggyback-based protocol

## 1 Introduction

Wireless sensor networks are composed of cost-effective wireless sensor nodes with the goal of reporting sensed data to the base station. Each wireless sensor node is equipped with a sensing module, processing module and radio module. The major applications of WSNs include health monitoring, industrial monitoring and automation, traffic surveillance and control, and environmental

monitoring [1]. WSNs are often deployed for critical applications and are required to transmit data reliably. The battery-powered sensor nodes are often deployed in harsh environments with limited energy harvesting opportunities in which the sensor node batteries cannot be replaced often. Considering that the sensor nodes have a limited battery capacity and are deployed in harsh environments, the sensor nodes must use their limited energy supply wisely. For the reliable and sustainable operation of WSNs, energy efficient techniques are required to prolong the network lifetime.

Because energy is the scarcest resource in WSNs, developing techniques to enhance the network lifetime has always been a major challenge for researchers. To prolong the network lifetime of WSNs, various energy-efficient techniques have been developed in the past few years. In

---

✉ Parmod Singh  
parmodsingh.cs01g@nctu.edu.tw  
Yaw-Chung Chen  
ycchen@cs.nctu.edu.tw

<sup>1</sup> Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, ROC

general, we can classify the energy efficient techniques used to preserve the limited energy of the sensor nodes into five categories [2], energy efficient routing (e.g., [3–5]), duty cycling methods (e.g., [6–8]), power control methods (e.g., [9, 10]), data reduction techniques (e.g., [11–13]) and node mobility (e.g., [14–16]). Along with the above-mentioned techniques, sensor nodes with energy harvesting modules that replenish their batteries have also proven to be effective in increasing their lifetime [17]. Among all the techniques used for energy conservation in WSNs, duty cycling has been considered to be one of the most effective [2, 18]. In this paper, we utilize the duty cycling technique that allows the sensor node to switch between the active and sleep modes to decrease the power consumption of the radio unit of the sensor node.

The power consumption of the radio module has been shown to be higher than that of the sensing and processing modules [7]. The radios used in WSNs basically operate in four distinct states: sleep, idle, receive and transmit. Among all these states, the radio consumes the most energy in the transmit and receive states. The energy consumed by a radio in the idle state is also significantly high, as the radio electronics keeps operating to detect packets, including the noise in the channel [2]. Therefore, it is not viable for energy-constrained wireless sensor nodes to have the radio remain in an active state. Thus, a sensor node must enter into the sleep state as much as possible whenever its services are not required by using effective duty cycling methods. The duty cycle is defined as the ratio of time during which the radio is in an active state (i.e., total active time) to the total time of the active and sleep states (total active time + total sleep time) [19]. The duty cycle of the sensor node is also dependent on the node density of the sensor network and is generally low for dense WSNs [19].

The duty-cycled medium access control (MAC) protocols are usually divided into two major categories: synchronous duty-cycled and asynchronous duty-cycled [6, 19] protocols. In synchronous duty-cycled WSNs, each node synchronizes with its neighboring nodes before transmitting the data through control messages (SYNC packets) to reduce the energy consumption, e.g., the S-MAC [20] and RMAC [21] protocols. The synchronous duty-cycled WSNs are considered efficient in communication and energy consumption, with an overhead in exchanging the duty cycle schedule with the neighboring nodes for synchronization [6, 19].

In asynchronous duty-cycled WSNs, each node turns on or off its radio following its own duty cycling schedule, which is independent of its neighboring nodes, e.g., X-MAC [22] and B-MAC [23] protocols. By achieving extremely low duty cycles, asynchronous duty-cycled MAC protocols have a lower energy consumption than the synchronous duty-cycled MAC protocols. Asynchronous duty-cycled MAC

protocols have no burden of control packet but have to find out the efficient way to communicate between the neighboring nodes [24]. Low power listening is one such technique that is employed by asynchronous duty-cycled MAC protocols. A sender node using the B-MAC protocol transmits a long preamble for a time equal to the sleep period of the receiver node. When the receiver node wakes up and detects the preamble, it stays awake to receive the data. The sender node starts transmitting the data after transmitting the long preamble. However, the downside of this approach is that some of the nodes may stay awake unnecessarily to receive the data meant for other nodes. The X-MAC protocol solves this problem by using many short preambles in place of one long preamble and includes the destination address in the preamble. With this approach, the X-MAC protocol allows the receiver nodes not involved in the communication to go to sleep immediately. The RI-MAC protocol [25] is a receiver-initiated asynchronous duty-cycled MAC protocol. In the RI-MAC protocol, the receiver node initiates the data transmission process by sending a beacon as an invitation for a transmission. A sender node waiting to send data then transmits the data after receiving the beacon from the intended receiver. After successfully receiving the data, the receiver node transmits an acknowledgment (ACK) packet that acts as an acknowledgment and a new invitation.

Multihop broadcast refers to the propagation of a broadcast packet such that every node in the network has a copy of it. Multihop broadcasting is one of the operations in wireless sensor networks that is required for various network services, such as route discovery, information dissemination or network-wide queries. In the synchronous sleep scheduling protocols, a broadcast is an easy operation because the synchronized nodes wake up with the same schedules. By using one broadcast transmission, more than one node can easily receive a broadcast packet. Moreover, the support for multihop broadcast transmission in synchronized sleep scheduling protocols can be achieved without much overhead. However, multihop broadcasting is not efficiently supported in many of the existing asynchronous duty cycling MAC protocols. As each node wakes up at different intervals in the asynchronous duty cycling MAC protocols according to its own independent duty cycle, all the neighboring nodes cannot be covered with a single broadcast transmission. To realize a broadcast procedure in asynchronous duty cycling MAC protocols, a sender node sends a unicast transmission to each of its neighbors independently. This method does not require any extra information, such as the broadcast progress or link quality information, and because of this, many redundant transmissions cause a significant amount of unnecessary energy consumption and increases the broadcast latency. Thus, designing an efficient multihop broadcast protocol for an asynchronous environment is a challenging but necessary task.

In this paper, we propose two energy efficient broadcast protocols, the Broadcast Progress-based Efficient Multihop Broadcast (BPEMB) and Tree-based Efficient Multihop Broadcast (TEMB) protocols, to realize multihop broadcast transmission in asynchronous duty cycling protocols without incurring any substantial overhead. Both the BPEMB and TEMB protocols are proactive multihop broadcast protocols because they require periodic exchanges of the link quality information and are thus suitable for networks with traffic bursts. The BPEMB and TEMB protocols achieve better performance in terms of energy consumption, redundant transmissions, broadcast packet cost and network load in both sparse and dense networks. The remainder of this paper is organized as follows: Sect. 2 summarizes the related work. In Sect. 3, a brief description of problem is given. Section 4 describes the motivation and network model. In Sect. 5, the first proposed broadcast protocol for asynchronous duty-cycled WSNs, the BPEMB protocol, is presented. Section 6 describes the TEMB protocol, the second proposed broadcast protocol for asynchronous duty-cycled WSNs. In Sect. 7, the simulation results are presented to demonstrate the effectiveness of our proposed protocols. Finally, Sect. 8 presents the conclusions of this study.

## 2 Related work

In recent years, protocols have been developed to efficiently support multihop broadcasts by exploiting information that is only available at the MAC layer. In the ADB protocol, the information about the progress of the broadcasts is efficiently distributed using the *ADB footer* that is included in the broadcast packet [26]. The ADB protocol's distribution of broadcast progress allows the nodes to transmit to only a subset of neighbors and go to sleep as soon as possible. However, the ADB protocol does not perform efficiently in polygonal topologies unless they are triangular. In the energy efficient network wide broadcast protocol for asynchronous wireless networks (EENWB) in [7], aggressive delegation is employed. When using aggressive cooperation among the neighboring sender nodes, some of the sender nodes sleep early at the cost of increasing the active time of the cooperating nodes [7]. The dynamic delegation-based efficient broadcast (DDEB) protocol dynamically handles the job of transmitting the broadcast packet to an uncovered node to the sender node who has the best link quality to the uncovered node so far [27]. The DDEB protocol avoids sending the broadcast packet over links with poor link quality and accelerates the broadcast process by taking advantage of the early covered neighbor.

The coverage efficiency-based broadcast protocol (CEB) for asynchronous WSNs is another multihop broadcast approach that not only considers the link quality and sleep/wake scheduling information of the neighbor nodes but also the coverage area in the broadcast process [28]. The CEB protocol performance is influenced by the order in which the neighboring nodes receive the broadcast packet. In the efficient multihop broadcasting protocol for asynchronous WSNs (EMBA), the authors use two techniques, specifically the forwarder's guidance and the overhearing of the broadcast messages and ACKs to efficiently broadcast packets [6]. The EMBA protocol uses the 1-hop and 2-hop neighbor link quality information for the delegation process. In the EMBA protocol, redundant transmissions are avoided through the guidance list, which is sent to each receiver node by the sender node. The guidance list provided by the sender node instructs how each receiver node should forward the broadcast message to its neighbor nodes.

Cheng et al. [29] presented a distributed minimum-delay energy-efficient flooding tree (MDET) protocol to construct an energy optimal tree with flooding delay bounding. In the MDET protocol, a shortest path tree is first constructed using the expected transmission count (ETX) metric, and then the tree is locally adjusted to improve the energy optimality with delay constraints. In [30], the receiver node postpones its wake-up time so that it can opportunistically overhear the broadcasting message sent by one of its neighbors. By using this opportunistic broadcasting transmission model, the nodes can save their energy with an increase in the average broadcast latency. Sinha et al. [31] presented an online throughput-optimal broadcast algorithm that uses packet-by-packet scheduling and makes routing decisions without constructing the network-wide spanning tree. In [32], the broadcast redundancy minimization scheduling (BRMS) scheme is presented that minimizes the number of broadcast transmissions. In the BRMS scheme, a centralized node has the active time slot information of each node in the network, which is used to construct the broadcast tree.

Yan et al. [33] presented an efficient multihop broadcasting with network coding (EMBNC) protocol to disseminate multiple packets. In the EMBNC approach, a forwarder only selects enough 2-hop neighbor nodes to accommodate the neighboring nodes as downstream forwarders, while the other 1-hop and 2-hop neighbor nodes act as auxiliary nodes. In the collision-tolerant scheduling (CTS) strategy, the critical nodes receive packets with reduced broadcast delays at the cost of collisions at the noncritical nodes [34]. In an adjustable duty cycle-based fast disseminate (ADCFD) scheme for smart wireless software-defined networks, the duty cycle of the nodes is adjusted to receive the data [35]. An opportunistic broadcasting transmission model that provides flexible control of the tradeoff between the average

broadcasting delay and the total energy consumption through a broadcasting cost function is presented in [36].

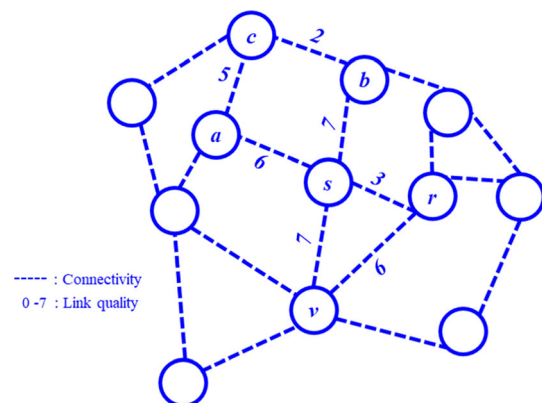
An efficient and fully distributed concurrent broadcast layer, called the chase, for flooding in asynchronous duty cycle networks is proposed in [37]. In the chase, a distributed random inter-preamble packet interval adjustment approach is used to meet the strict time and signal strength requirements for concurrent broadcasts. Yan et al. [38] presented a scheme that utilizes network coding to maximize the benefit of overhearing the multiple neighbor nodes of a receiver. In an enhanced version of the efficient context-aware multihop broadcasting (E-ECAB) protocol, the authors combine the duty cycle mechanism and a context-aware paradigm to optimize the resource usage and satisfy the application requirements [39]. In an adaption broadcast radius-based code dissemination scheme, a larger broadcast radius is set in the areas with more energy left so that the program codes can reach the edge of the network from the source in fewer hops, which decreases the number of broadcasts and delay [40]. In [41], the authors investigated the broadcast problem in multichannel duty-cycled wireless body area networks and devised a 2-D schedule to specify the channel hopping and wake-up time slot selection rules. In a neighbor knowledge-based broadcast scheme, *hello messages* are exchanged to determine the neighbor density, the ratio, and the number of one-hop uncovered neighbors, upon which the rebroadcast probability and delay are adjusted adaptively [42].

The X-MAC-UPMA protocol [43] is the X-MAC implementation for a unified power management architecture (UPMA). In the X-MAC-UPMA protocol, the broadcast process is realized by repeatedly sending duplicate copies of the broadcast packets over the sleep interval. The RI-MAC protocol for dynamic traffic loads in WSNs supports broadcast in two ways [25]. One way, which is similar to that of the X-MAC-UPMA protocol, is to repeatedly transmit the data packet back-to-back throughout the sleep interval. Another way is to have the sender node send the data packet as a unicast transmission to each of its 1-hop neighbors. Zhang et al. [44] proposed a probabilistic rebroadcast scheme that is based on neighbor coverage knowledge includes an additional coverage ratio and a connectivity factor for reducing the routing overhead in mobile ad hoc networks. In the probabilistic rebroadcast protocol, the rebroadcast delay is dynamically calculated. The rebroadcast delay is then used to determine the forwarding order to effectively exploit the neighbor coverage knowledge. However, in asynchronous WSNs, due to the independent sleep/wake schedules of the nodes, the coverage problem becomes more complicated.

### 3 Problem description

In asynchronous duty cycling environments, each node has a different wakeup time with different schedules. Thus, it is not possible for a sender node to send a broadcast packet to all of its neighboring nodes in one broadcast transmission. For example, in Fig. 1, Node  $s$  is a source node of a broadcast packet. For a receiver-initiated MAC protocol, upon receiving a beacon from node  $v$ , node  $s$  unicasts the transmission of broadcast packet to node  $v$ . Now, node  $s$  and  $v$  stay awake to transmit the broadcast packet to their common neighbor, node  $r$ . Upon receiving the beacon from node  $r$ , both the nodes simultaneously transmit the broadcast packet to node  $r$ , leading to a collision at  $r$ . Each of the nodes,  $s$  and  $v$ , then retransmit the collided message after a backoff interval. Now, node  $r$  unnecessarily receives two redundant broadcast messages after the collision. In another scenario with a quadrangular topology of nodes  $s$ ,  $a$ ,  $b$  and  $c$ , node  $s$  unicasts a transmission to nodes  $a$  and  $b$ . After receiving the broadcast packet, both nodes  $a$  and  $b$  stay awake to deliver the broadcast packet to their common neighbor, node  $c$ . Without any multihop broadcast policy, nodes  $a$  and  $b$  both deliver the broadcast packet to their common neighbor, node  $c$ , causing redundant transmissions and collisions at node  $c$ . However, these redundant transmissions and collisions can be avoided effectively with cooperation among the neighbors such that the node with the best link quality to the receiver node takes responsibility to forward the packet to the receiver node [6, 7].

For preamble-based MAC sender-initiated protocols, a sender node can decide in what order the broadcast packet should be transmitted. Several policies are available, such as sending the broadcast packets randomly to neighbors, in the relative order of the wakeup times of neighbors, according to the neighbor IDs, and according to the link quality order. However, in this case, any transmitting



**Fig. 1** An example of a WSN consisting of heterogeneous local topologies with triangles and quadrangles

policy will result in redundant transmissions and collisions when some of the common neighbors of the node try to transmit the broadcast packet at the same time. Redundant transmissions and collisions lead to significant energy dissipation and broadcast delay. The broadcast packets must also be resent if they are lost due to poor link quality. Moreover, some sender nodes remain unnecessarily awake. For example, in Fig. 1, nodes  $s$  and  $v$  both stay awake to forward the broadcast packet to their common neighbor, node  $r$ . However, by using an efficient broadcast policy, node  $s$  can go to sleep and let node  $v$  transmit the broadcast packet to node  $r$  because it has better link quality to node  $r$ . Therefore, it is an important task to design an energy efficient multihop broadcast protocol that can utilize the link quality effectively to avoid redundant transmissions and minimize the active time of the sender nodes.

## 4 Motivation and network model

### 4.1 Motivation

In past research, it has been shown that in randomly deployed WSNs, the total ratio of 3-gons (triangles) to 4-gons (quadrangles) is more than 90% in both sparse and dense networks [6]. In our work, we take advantage of this fact in designing the BPEMB and TEMB protocols to reduce the number of redundant transmissions and collisions. As it is unavoidable to have both triangles and quadrangles in a network topology, avoiding redundant transmissions and collisions in such a network has a major impact on the overall performance of the multihop broadcast protocol.

Our goal is to design an efficient multihop broadcast protocol with a simple design without extra complexity (e.g., guidance lists) that can achieve better overall performance. Additionally, in our work, the sender node takes an independent decision about which nodes it will unicast a broadcast packet to and which nodes it will not cover. In the related works on designing energy efficient multihop broadcast protocols for asynchronous duty-cycled wireless sensor networks, the basic design is based on a guidance list that is sent to the receiver node by the sender node. The guidance list, which is piggybacked in the broadcast packet, instructs the receiver node about which nodes in its 1-hop neighborhood to unicast the broadcast packet to. In the BPEMB and TEMB protocols, the decision of forwarding the broadcast packet is not influenced by the guidance list of any other node except the node itself.

### 4.2 Network model

We consider a multihop WSN consisting of stationary nodes with no ability to move. Each node in the network

has a unique node ID and the same communication range. To avoid retransmissions, the broadcast packet must be transmitted over the best links; otherwise, retransmissions will cause additional power consumption. Both the BPEMB and TEMB algorithms use the link quality information that can be easily provided from existing link estimation techniques such as ETX [45], Four-bit link estimation [46], and F-LQE [47]. We assume that the links are symmetrical. In this study, the link quality is divided into 8 levels (0–7), where 7 is the best link quality and 0 is the worst. Table 1 lists the notation used in this paper.

Each node maintains two tables, a 1-hop neighbor table and a 2-hop neighbor table. A node's 1-hop neighbor table consists of the node IDs and link quality of its 1-hop neighbor nodes. Node  $n$  constructs a *hello message* that includes sequenced pairs of  $N_n^{1-nbrs}$  and  $LQ(n, m)$ . Node  $n$  then sends this *hello message* to each of its 1-hop neighbors to exchange the link quality information. We call this whole operation the *hello procedure*. The *hello procedure* is executed at regular intervals so that each node has the latest link quality information. We assume that the link quality information remains valid until a new *hello message* is received and the link quality is updated. Each node also maintains a 2-hop neighbor table that includes the 2-hop neighbor list and the link quality information received through the *hello procedure*. Each node in a network of  $n$  nodes can have knowledge of its 2-hop neighbor nodes with the total of  $O(n)$  messages [48].

A node is said to be covered if it has already received the broadcast packet or will eventually receive it. The forwarding is the node that generates the broadcast packet or the node that forwards a broadcast packet generated by the source node. When a node  $u$  receives a broadcast message from node  $v$ , then node  $v$  is referred to as a  $BPKT_i.sender$ .

## 5 BPEMB protocol

In this section, the first proposed energy efficient broadcast protocol, the BPEMB protocol, is described. In BPEMB protocol, each sender node makes its forwarding decision based on its own judgment using the broadcast progress and link quality information. A node sends the broadcast progress in the  $BPKT_i.lastHopCoveredNodes$  and  $BPKT_i.last2HopCoveredNodes$  lists by piggybacking it in the broadcast packet. Both of these lists contain the node ID's of the nodes that have already received the broadcast packet.

In the BPEMB protocol, set  $S_s^{ucv}(i)$  contains all the nodes which the forwarding node  $s$  considers in deciding whether the nodes in  $S_s^{ucv}(i)$  should be delegated or if it

**Table 1** Summary of notations

Notation	Description
$N_n^{1-nbrs}$	1-hop neighbor set of node $n$ along with link quality information
$BPKT_i$	Broadcast packet $i$ , also referred to as broadcast message $i$
$S_n^{cv}(i)$	Set of neighboring nodes that have already received $BPKT_i$ or will receive $BPKT_i$
$S_n^{ucv}(i)$	Set of neighboring nodes of node $n$ that have not received $BPKT_i$
$S_n^{tx}(i)$	Set of neighboring nodes to which node $n$ will transmit $BPKT_i$
$BPKT_i.lastHopCoveredNodes$	A list of nodes to which a 1-hop upstream neighbor of a node has already transmitted the broadcast packet
$BPKT_i.last2HopCoveredNodes$	A list of nodes to which a 2-hop upstream neighbor of a node has already transmitted the broadcast packet
$LQ(n, m)$	Link quality between nodes $n$ and $m$

should take the responsibility itself to transmit the broadcast packet to them. The forwarding node  $s$  chooses each node from set  $S_s^{ucv}(i)$  in descending order of link quality ( $LQ(s, v) \forall v \in S_s^{ucv}(i)$ ) from  $S_s^{ucv}(i)$  and takes the required decision. If the forwarding node  $s$  has the best link quality to the neighboring node,  $v \in S_s^{ucv}(i)$ , of any of the other neighbors of node  $v$ , i.e.,  $LQ(s, v) > LQ(j, v) \forall j \in N_v^{1-nbrs}$ , then node  $s$  transmits the broadcast packet to node  $v$ . However, if there is some neighbor of node  $v$  that has a better link quality to  $v$  than node  $s$ 's link quality to node  $v$ , then node  $s$  either delegates node  $v$  or forwards the broadcast message to node  $v$ .

Forwarding node  $s$  stores any node  $k \in N_v^{1-nbrs}$  in set  $S_v^{bestLQ}(i)$  if it satisfy the condition,  $LQ(s, v) < LQ(k, v)$  and sorts  $S_v^{bestLQ}(i)$  in descending order of link quality,  $LQ(k, v) \forall k \in N_v^{1-nbrs}$ . Suppose that node  $t \in S_v^{bestLQ}(i)$ , before the forwarding node  $s$  delegates node  $v$  to node  $t$ , it has to check whether node  $t$  has already received the broadcast packet or will eventually receive it. First, node  $s$  checks for the existence of node  $t$  in set  $S_s^{cv}(i)$ , if it exists in set  $S_s^{cv}(i)$ , then node  $s$  delegates the responsibility of covering node  $v$  to node  $t$ . If node  $t$  does not exist in set  $S_s^{cv}(i)$ , then node  $s$  checks whether node  $t$  exists as a 1-hop neighbor of any node in  $S_s^{cv}(i)$ . The sender node  $s$  may not have the 1-hop neighbor information of all the nodes in  $S_s^{cv}(i)$  because some of the nodes in  $S_s^{cv}(i)$  may not be 1-hop neighbors of the sender node  $s$ . Let us say that node  $t$  is a 1-hop neighbor of node  $k \in S_s^{cv}(i)$ , then node  $s$  needs to check whether node  $t$  will eventually receive the broadcast packet from node  $k$  with better link quality or not. If  $LQ(s, v) + LQ(v, t) < LQ(t, v) + LQ(k, t)$ , then node  $s$  delegates the responsibility to cover node  $v$  to node  $t$ . If  $LQ(s, v) + LQ(v, t) > LQ(t, v) + LQ(k, t)$ , then node  $s$  forwards the broadcast packet to node  $v$ . Node  $v$  can be covered by node  $t$ , but in that case, node  $t$  will receive the broadcast packet from node  $k$  by using a link with poor link quality. Clearly, the BPEMB protocol avoids transmissions over poor

quality links. The pseudo code of the BPEMB protocol is depicted in Algorithm 1.

The sender node executes the BPEMB protocol for every new broadcast packet that it receives. In lines 2–12 of Algorithm 1, the BPEMB protocol initializes the required node sets namely,  $S_s^{ucv}(i)$ ,  $S_s^{cv}(i)$ , and  $S_s^{tx}(i)$ . All the nodes that have already received broadcast packet  $BPKT_i$ , before it has reached the current node  $s$  are removed from its uncovered nodes in set  $S_s^{ucv}(i)$ . The nodes that have already been covered are added to the covered node set,  $S_s^{cv}(i)$ . In lines 14–59 of Algorithm 1, the sender node either takes the responsibility to forward the broadcast packet to the neighboring node or the responsibility is delegated to another node. The process of determining whether to delegate the responsibility of forwarding the broadcast packet to another node also ensures that the broadcast packet will ultimately reach the delegated node. To ensure that the delegated node receives the broadcast packet, the sender node  $s$  also checks that the node that will cover the delegated node has already received  $BPKT_i$  or will receive it. In lines 17–22 of Algorithm 1, sender node  $s$  checks whether its own link quality to node  $v \in S_s^{ucv}(i)$  is better than that of any other 1-hop neighbor node of  $v$ . If the link quality of the sender node  $s$  is better than that of all of the other 1-hop neighbor nodes of node  $v$ , i.e.,  $LQ(s, v) > LQ(j, v) \forall j \in N_v^{1-nbrs}$ , then node  $v$  is covered by node  $s$  (lines 55–57). If there is a 1-hop neighbor of node  $v$  with better link quality to node  $v$  than that of the sender node, then all those 1-hop neighbor nodes are put into set  $S_v^{bestLQ}(i)$  (line 20). Before the sender node delegates node  $v$  to the best node  $t \in S_v^{bestLQ}(i)$  from set  $S_v^{bestLQ}(i)$ , it has to make sure that node  $t$  has a copy of the broadcast packet or the sender node will receive it. In line 24 of Algorithm 1, all the nodes in  $S_v^{bestLQ}(i)$  are sorted by decreasing order of the link quality,  $LQ(t, v) \forall t \in S_v^{bestLQ}(i)$ . This ensures that, out of all the nodes that can cover node  $v$ , the node with the best link quality takes the responsibility of sending the broadcast packet to node  $v$ .

**Algorithm 1** BPEMB Protocol

---

```

1: //  $s$ : sender,  $v$ : receiver;  $i$ : broadcast message  $i$ 
2:  $S_s^{ucv}(i) \leftarrow N_v^{l-nbrs} - BPKT_i.sender$ 
3:  $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{BPKT_i.sender\}$ 
4:  $S_s^{tx}(i) \leftarrow \emptyset$ 
5: for each node  $r \in BPKT_i.lastHopCoveredNodes$ 
6:    $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{r\}$ 
7:    $S_s^{ucv}(i) \leftarrow S_s^{ucv}(i) - \{r\}$ 
8: end for
9: for each node  $r \in BPKT_i.last2HopCoveredNodes$ 
10:   $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{r\}$ 
11:   $S_s^{ucv}(i) \leftarrow S_s^{ucv}(i) - \{r\}$ 
12: end for
13: sort all nodes in  $S_s^{ucv}(i)$  in descending order of link quality  $LQ(s, v) \forall v \in S_s^{ucv}(i)$ 
14: for each node  $v \in S_s^{ucv}(i)$ 
15:   $S_v^{bestLQ}(i) \leftarrow \emptyset$ 
16:  bool meBestNbr = true
17:  for each node  $j \in N_v^{l-nbrs}$ 
18:    if  $LQ(s, v) < LQ(j, v)$  then
19:      MeBestNbr = false
20:       $S_v^{bestLQ}(i) \leftarrow S_v^{bestLQ}(i) \cup \{j\}$ 
21:    end if
22:  end for
23: if meBestNbr = false then
24:  sort all nodes in  $S_v^{bestLQ}(i)$  in descending order of link quality  $LQ(j, v) \forall j \in S_v^{bestLQ}(i)$ 
25:  bool coveredBySomeNode = false
26:  for each node  $t \in S_v^{bestLQ}(i)$ 
27:    if  $t \in S_s^{cv}(i)$  then
28:       $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{v\}$ 
29:       $S_s^{ucv}(i) \leftarrow S_s^{ucv}(i) - \{v\}$ 
30:      coveredBySomeNode = true
31:    break;
32:    end if
33:  end for
34:  if coveredBySomeNode = false then
35:    bool delegateFound = false
36:    for each node  $t \in S_v^{bestLQ}(i)$ 
37:      for each node  $k \in S_s^{cv}(i)$ 
38:        if  $t \in N_k^{l-nbrs}$  then
39:          if  $LQ(s, v) + LQ(v, t) < LQ(t, v) + LQ(k, t)$  then
40:            delegateFound = true
41:          end if
42:        end if
43:      end for
44:    end for
45:  end if
46:  if delegateFound = true then
47:     $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{v\}$ 
48:     $S_s^{ucv}(i) \leftarrow S_s^{ucv}(i) - \{v\}$ 
49:  else if delegateFound = false then
50:     $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{v\}$ 
51:     $S_s^{tx}(i) \leftarrow S_s^{tx}(i) \cup \{v\}$ 
52:     $S_s^{ucv}(i) \leftarrow S_s^{ucv}(i) - \{v\}$ 
53:  end if
54: else
55:   $S_s^{cv}(i) \leftarrow S_s^{cv}(i) \cup \{v\}$ 
56:   $S_s^{tx}(i) \leftarrow S_s^{tx}(i) \cup \{v\}$ 
57:   $S_s^{ucv}(i) \leftarrow S_s^{ucv}(i) - \{v\}$ 
58: end if
59: end for
60:  $BPKT_i.last2HopCoveredNodes \leftarrow BPKT_i.lastHopCoveredNodes$ 
61:  $BPKT_i.lastHopCoveredNodes \leftarrow S_s^{tx}(i)$ 

```

---

In lines 26–33 of Algorithm 1, node  $s$  checks whether node  $t$  has already received the broadcast packet or will receive it. Node  $s$  checks for the existence of each node  $t \in S_v^{\text{bestLQ}}(i)$  in  $S_s^{cv}(i)$ . If node  $t$  exists in set  $S_s^{cv}(i)$ , then node  $t$  is the best node to cover node  $v$  among all the other nodes in set  $S_v^{\text{bestLQ}}(i)$ .

If none of the nodes in set  $S_v^{\text{bestLQ}}(i)$  exists in set  $S_s^{cv}(i)$ , In lines 34–45 of Algorithm 1, node  $s$  checks for the existence of node  $t$  in the 1-hop neighbor set of the nodes that are in covered set  $S_s^{cv}(i)$  (if a 1-hop neighbor set of a node in  $S_s^{cv}(i)$  is available). Here, the sender node  $s$  also does one additional necessary check; if node  $t \in N_k^{1-\text{nbrs}}$  ( $k \in S_s^{cv}(i)$ ), the sender node  $s$  ensures that node  $t$  will also be covered by the node with best link quality to it. If node  $s$  can find any node  $k \in S_s^{cv}(i)$  such that  $t \in N_k^{1-\text{nbrs}}$  and  $LQ(s, v) + LQ(v, t) < LQ(t, v) + LQ(k, t)$ , it delegates the responsibility of covering node  $v$  to node  $t$  (line 46–48). If no such node is found, then node  $s$  covers node  $v$  (line 49–53).

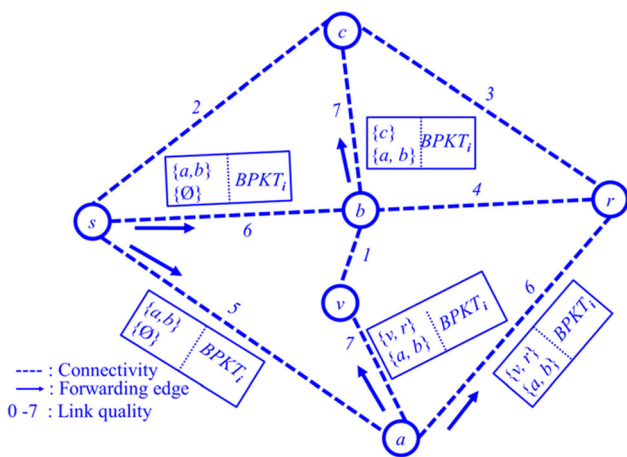
Finally, in lines 60–61, the sender node  $s$ , puts all the nodes in the  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  list into the  $\text{BPKT}_i.\text{last2HopCoveredNodes}$  list. All the nodes in set  $S_s^{tx}(i)$  are transferred to the  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  list. Both the  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  and  $\text{BPKT}_i.\text{last2HopCoveredNodes}$  lists are piggybacked with the broadcast packet to all the nodes in  $S_s^{tx}(i)$ . Both lists, the  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  list and the  $\text{BPKT}_i.\text{last2HopCoveredNodes}$  list, provide the necessary information to the receiving node about the up-to-date broadcast progress.

We illustrate the overall process of the BPEMB protocol in a WSN with a complex quadrangle topology, as shown in Fig. 2. Node  $s$  is the generating node of the broadcast

packet,  $\text{BPKT}_i$ . Node  $s$  initializes its sets as  $S_s^{ucv}(i) = \{b, a, c\}$ ,  $S_s^{cv}(i) = \{\emptyset\}$ , and  $S_s^{tx}(i) = \{\emptyset\}$ . Node  $s$  first selects node  $b$  from its uncovered set,  $S_s^{ucv}(i)$ , to check whether it should take responsibility for sending the broadcast packet  $\text{BPKT}_i$ , or let some another node cover it. Out of all the neighbors of node  $b$ , there exists one 1-hop neighbor “node  $c$ ” of node  $b$  that has better link quality to node  $b$  than that of node  $s$ , i.e.,  $LQ(s, b) < LQ(c, b)$ . Before node  $s$  delegates the responsibility of broadcasting  $\text{BPKT}_i$  to node  $c$ , it has to make certain that node  $c$  has either received  $\text{BPKT}_i$  or will eventually receive it. Since node  $c$  does not exist in  $S_s^{cv}(i)$  and is not a 1-hop neighbor of any node present in  $S_s^{cv}(i)$  ( $S_s^{cv}(i)$  is empty at this point of time), node  $s$  takes responsibility to send  $\text{BPKT}_i$  to node  $b$  and updates its uncovered set, covered set and  $S_s^{tx}(i)$  set as  $S_s^{ucv}(i) = \{a, c\}$ ,  $S_s^{cv}(i) = \{b\}$  and  $S_s^{tx}(i) = \{b\}$ . Next, node  $s$  finds that node  $a$  has two neighbors,  $v$  and  $r$ , with better link quality than its own to node  $a$ . Nodes  $v$  and  $r$  are both entered into set  $S_a^{\text{bestLQ}}(i)$ , and  $S_a^{\text{bestLQ}}(i)$  is sorted in decreasing order based on the link quality of the nodes in it to node  $a$  ( $S_a^{\text{bestLQ}}(i) = \{v, r\}$ ). Now, node  $s$  checks whether any of the nodes in  $S_a^{\text{bestLQ}}(i)$  exist in  $S_s^{cv}(i)$ . None of the nodes in  $S_a^{\text{bestLQ}}(i)$  exist in set  $S_s^{cv}(i)$ , so node  $s$  determines whether any node in  $S_a^{\text{bestLQ}}(i)$  is a 1-hop neighbor of any node present in  $S_s^{cv}(i)$ . Both nodes  $v$  and  $r$  are 1-hop neighbors of node  $b$ , but these nodes fail the conditions,  $LQ(s, a) + LQ(a, v) < LQ(b, v) + LQ(v, a)$  and  $LQ(s, a) + LQ(a, r) < LQ(b, r) + LQ(r, a)$  respectively. Therefore, node  $s$  takes the responsibility of broadcasting  $\text{BPKT}_i$  to node  $a$  and updates the sets as  $S_s^{ucv}(i) = \{c\}$ ,  $S_s^{cv}(i) = \{b, a\}$  and  $S_s^{tx}(i) = \{b, a\}$ . Finally, node  $s$  checks for the responsibility of broadcasting  $\text{BPKT}_i$  for the remaining node  $c$  in  $S_s^{ucv}(i)$ . The link quality of node  $c$ ’s 1-hop neighbor,  $b$ , is better that of node  $s$  to  $c$ , and node  $b$  is in set  $S_s^{cv}(i)$ , i.e.,  $LQ(s, c) < LQ(b, c)$  and  $b \in S_s^{cv}(i)$ . Thus, sender node  $s$  delegates the responsibility of broadcasting the  $\text{BPKT}_i$  to node  $b$ . Sender node  $s$  broadcasts the  $\text{BPKT}_i$  packet to the nodes in  $S_s^{tx}(i) = \{b, a\}$ . Node  $s$  has avoided sending  $\text{BPKT}_i$  to node  $c$  as the probability of losing  $\text{BPKT}_i$  is higher due to the bad link quality.

Node  $b$  updates its covered and uncovered sets as  $S_b^{cv}(i) = \{s, a, b\}$  and  $S_b^{ucv}(i) = \{c, r, v\}$  from the  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  and  $\text{BPKT}_i.\text{last2HopCoveredNodes}$  lists piggybacked in the  $\text{BPKT}_i$ . Node  $b$  transmits  $\text{BPKT}_i$  to node  $c$  as the link quality between node  $b$  and  $c$  is the best among all the other 1-hop neighbors of node  $c$ . Both nodes  $r$  and  $v$  are best covered by node  $a$  and node  $a$  exists in  $S_b^{cv}(i)$ , so node  $b$  let node  $a$  cover nodes  $v$  and  $r$ .

Node  $a$  after receiving the broadcast packet from source node  $s$  updates its covered and uncovered set as  $S_a^{cv}(i) = \{s, a, b\}$  and  $S_a^{ucv}(i) = \{v, r\}$ . Node  $a$  has the best link quality to



**Fig. 2** Snapshot of the broadcast packet including the piggybacked  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  and  $\text{BPKT}_i.\text{last2HopCoveredNodes}$  lists. The direction of the flow of the broadcast packet is shown by the forwarding edge



send the broadcast packet to nodes  $v$  and  $r$ , so node  $a$  unicasts  $\text{BPKT}_i$  to nodes  $v$  and  $r$ .

The uncovered set  $S_c^{\text{ucv}}(i)$  of node  $c$  is empty after updating it, therefore node  $c$  does not participate in broadcasting  $\text{BPKT}_i$  to any of its neighbors. Node  $v$  has a similar situation for  $\text{BPKT}_i$  as node  $c$ . Both nodes  $b$  and  $c$  give up responsibility to transmit  $\text{BPKT}_i$  to node  $r$  thus avoid redundant transmissions to node  $r$  and possible collisions at node  $r$ . Node  $r$  has node  $c$  in its uncovered set  $S_r^{\text{ucv}}(i)$ , following its update of  $S_r^{\text{ucv}}(i)$  after receiving  $\text{BPKT}_i$  from node  $a$ . Node  $r$  does not transmit  $\text{BPKT}_i$  to node  $c$  after finding that node  $c$  is best covered by node  $b$  and node  $b$  exists in its covered set  $S_r^{\text{cv}}(i)$ . By not transmitting  $\text{BPKT}_i$  to node  $c$ , the BPEMB protocol successfully avoids the redundant transmission of  $\text{BPKT}_i$  to node  $c$ . In this particular example, a broadcast using the BPEMB protocol takes a total of 5 unicast transmissions to propagate the broadcast packet to each node in the network.

### 5.1 Handling overhearing of the broadcast messages and ACKs

The probability that a node can overhear the broadcast message or ACK destined for another node in a WSN is quite high. The BPEMB protocol exploits this opportunity to reduce the number of transmissions. If the forwarding node  $s$ , overhears the  $\text{BPKT}_i$  transmitted by node  $u$  to node  $v$  during an active state of its duty cycle, it will update sets  $S_s^{\text{ucv}}(i)$  and  $S_s^{\text{cv}}(i)$  as follows:

$$S_s^{\text{ucv}}(i) \leftarrow S_s^{\text{ucv}}(i) - \{u\} \quad (1)$$

$$S_s^{\text{cv}}(i) \leftarrow S_s^{\text{cv}}(i) \cup \{u\} \quad (2)$$

An overhearing of an  $\text{ACK}_i$  indicates that both the sender and receiver nodes of the  $\text{ACK}_i$  have successfully received  $\text{BPKT}_i$ . In this case, forwarding node  $s$  deletes the node IDs of nodes  $v$  and  $u$  from its uncovered set,  $S_s^{\text{ucv}}(i)$ . If the forwarding node  $s$  overhears the  $\text{ACK}_i$  transmitted by node  $u$  to node  $v$  during an active state of its duty cycle, it will update sets  $S_s^{\text{ucv}}(i)$  and  $S_s^{\text{cv}}(i)$  as follows:

$$S_s^{\text{ucv}}(i) \leftarrow S_s^{\text{ucv}}(i) - \{v, u\} \quad (3)$$

$$S_s^{\text{cv}}(i) \leftarrow S_s^{\text{cv}}(i) \cup \{v, u\} \quad (4)$$

With this simple addition to the BPEMB protocol, the number of transmissions required to cover the 1-hop neighbors of the node decreases significantly. Since, the number of transmissions decreases, the active time of the sender node also decreases, which in turn saves the energy of the sender node.

### 5.2 Handling of network failure in the BPEMB protocol

Messages can be lost in wireless networks due to collision or link failure. The BPEMB protocol uses the network failure mechanism that is incorporated within the underlying asynchronous duty-cycled MAC protocol. A forwarding node  $s$  will simply retransmit the  $\text{BPKT}_i$  after a retransmit timeout in case of a lost  $\text{BPKT}_i$  or  $\text{ACK}_i$ . If the receiver node doesn't respond within the maximum number of allowed retransmission attempt, BPEMB protocol takes the decision according to the underlying asynchronous duty cycled MAC protocol, i.e., the broadcast packet is dropped. However, the handling of a  $\text{BPKT}_i$  or  $\text{ACK}_i$  in overhearing case is a complex task. Forwarding node  $s$  has overheard the  $\text{ACK}_i$  sent by node  $u$  to node  $v$  and it has been lost. As IDs of node  $v$  and  $u$  has been removed from the uncovered set  $S_s^{\text{ucv}}(i)$  by the forwarding node  $s$  but node  $v$  hasn't received the  $\text{ACK}_i$  yet. Since,  $v$  hasn't received the  $\text{ACK}_i$  within retransmit timeout, so it will re-send  $\text{BPKT}_i$  to node  $u$ . Another way to solve this problem is to implement a process similar to one that is mentioned in the EMBA protocol. Each node saves the addresses of the sender and receiver nodes and the broadcast number  $i$ , of the recently overheard  $\text{BPKT}_i$  or  $\text{ACK}_i$  for a period of  $n$  duty cycles. After waiting for this duration, the forwarding node  $s$  updates its sets as mentioned in Sect. 5.1.

### 6 TEMB protocol

To design an energy efficient BPEMB protocol for asynchronous duty-cycled MAC protocols, first, we utilize the best quality links to transfer the data. Second, we try to send the broadcast packets to each node in the network using the minimum number of unicast transmissions. To send a broadcast packet to each node in an error-free network of  $m$  nodes, a total of  $m - 1$  unicast transmissions are required.

The TEMB protocol realizes the above two properties of the BPEMB protocol through a simple design. In the TEMB protocol, the broadcast packets do not have to piggyback the two lists that are used in the BPEMB protocol, namely  $\text{BPKT}_i.\text{lastHopCoveredNodes}$  and  $\text{BPKT}_i.2\text{lastHopCoveredNodes}$ . In the TEMB protocol, each node constructs a maximum spanning tree (MST) based on the partial information of the network available to it. A maximum spanning tree of an undirected weighted graph is the spanning tree with the maximum weight. A maximum spanning tree can be easily computed by calculating the reciprocal of the weights and applying the Kruskal

algorithm. Kruskal algorithm is an algorithm in graph theory that is used to find the MST of a connected weighted graph. We use the Kruskal algorithm in the TEMB protocol to obtain a MST at each node. A maximum spanning tree  $T$ , constructed by node  $n$  includes all of its 1-hop and 2-hop neighbors, where the total link quality of all the edges connecting the nodes are maximized. If node  $n$  has an edge  $(n, v) \in T$  and  $v \neq \text{BPKT}_i.\text{sender}$ , then node  $n$  transmits the broadcast packet to node  $v$ . The pseudocode of the TEMB protocol is presented in Algorithm 2.

```

Algorithm 2 TEMB Protocol
1: //  $s$ : sender,  $v$ : receiver;  $i$ : broadcast message  $i$ 
2:  $S_s^{\text{tx}}(i) \leftarrow \emptyset$ 
3:  $\text{SenderNode} \leftarrow \text{BPKT}_i.\text{sender}$ 
4: Construct a graph  $g$  using 1-hop and 2-hop neighbors information
5:  $T \leftarrow$  get the spanning tree of graph  $g$ 
6: for each edge  $(u, v) \in T$ 
7:   if  $u = s$  then
8:     if  $v \neq \text{SenderNode}$  then
9:        $S_s^{\text{tx}}(i) \leftarrow S_s^{\text{tx}}(i) \cup \{v\}$ 
10:    end if
11:  end if
12: end for
    
```

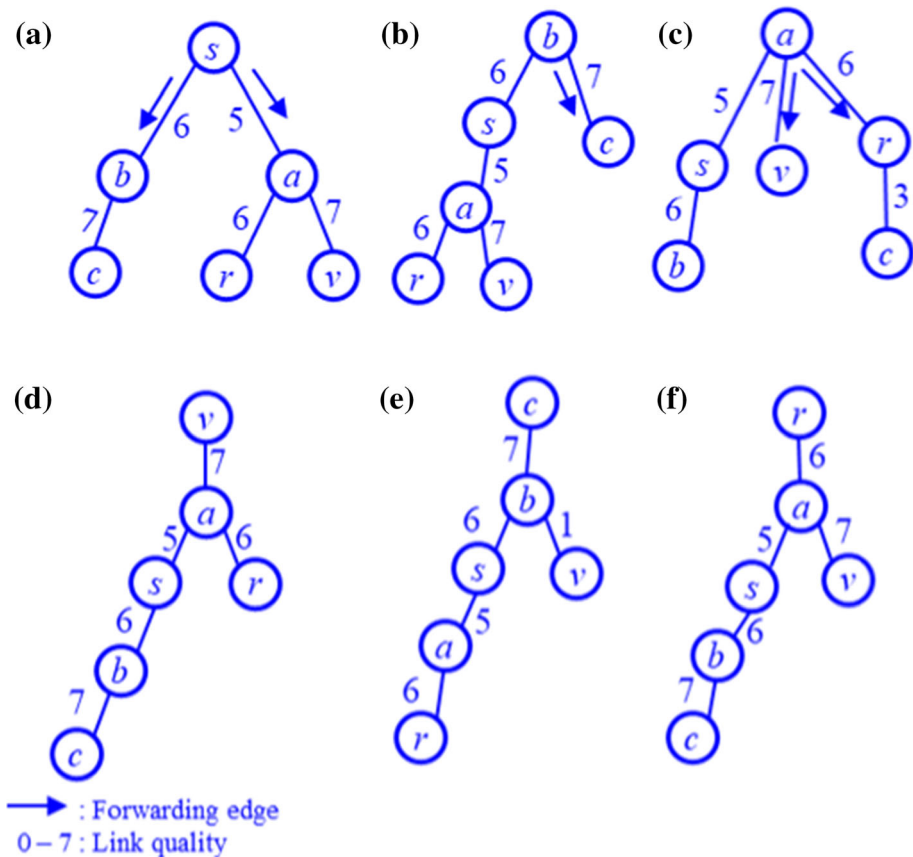
The SenderNode variable stores the ID of the node from which the forwarding node  $s$  receives the broadcast packet

$\text{BPKT}_i$ . In line 4 of Algorithm 2, the forwarding node  $s$  constructs a weighted graph  $g$ , using the 1-hop and 2-hop neighbor information available to it through the *hello procedure*. Each vertex in  $g$  is essentially a node ID, and the weight on edge  $(u, v)$  of  $g$  is the link quality between nodes  $u$  and  $v$ . In line 5, an MST  $T$  of  $g$  is found. In lines 6–12 of Algorithm 2, the forwarding node  $s$  compares vertex  $u$  of each edge  $(u, v) \in T$  with its own ID, if true, then it compares vertex  $v$  of edge  $(u, v)$  with the SenderNode variable. If the edge’s vertex  $v$  does not match the SenderNode variable, then the forwarding node  $s$ , puts  $v$  into  $S_s^{\text{tx}}(i)$  (lines 8–10). The forwarding node,  $s$ , broadcasts  $\text{BPKT}_i$  to each node in  $S_s^{\text{tx}}(i)$ .

Next, through an example, the working of the TEMB protocol is explained. The network shown in Fig. 1 is considered. Forwarding node  $s$  constructs an MST from its 1-hop and 2-hop neighbor information, as shown in Fig. 3(a). There are two nodes,  $b$  and  $a$ , with which  $s$  forms an edge in the MST tree. Nodes  $b$  and  $a$  are put into the  $S_s^{\text{tx}}(i)$  set, and node  $s$  transmits a broadcast packet to both of these nodes.

After receiving a broadcast packet from node  $s$ , both nodes  $b$  and  $a$  form an MST, as shown in Fig. 3(b), (c), respectively. Node  $b$ ’s MST has two nodes,  $s$  and  $c$ , that

**Fig. 3** Example of the TEMB protocol. The direction of the flow of the broadcast packet is shown by the forwarding edge. An edge between nodes that are included in the MST is shown by a line. **a** MST formed by node  $s$ . **b** MST formed by node  $b$ . **c** MST formed by node  $a$ . **d** MST formed by node  $v$ . **e** MST formed by node  $c$ . **f** MST formed by node  $r$



form an edge with it. As node  $b$  has received the  $BPKT_i$  from node  $s$ , it will not transmit  $BPKT_i$  to it. Thus, node  $b$  transmits the  $BPKT_i$  to node  $c$ . Node  $a$ 's MST has three edges connected to it with nodes  $s$ ,  $v$  and  $r$ . Node  $a$  transmits  $BPKT_i$  to node  $v$  and  $r$ . Node  $a$  doesn't forwards  $BPKT_i$  to node  $s$  as it has received the  $BPKT_i$  from node  $s$ .

Nodes  $v$ ,  $c$  and  $r$  form an MST after each receiving  $BPKT_i$  from their respective  $BPKT_i.sender$ , as shown in Fig. 3(d)–(f), respectively. However, none of these nodes,  $c$ ,  $v$  or  $r$  transmits  $BPKT_i$ . These nodes form only one edge with the node from which they have received  $BPKT_i$ . Therefore, none of these nodes have to broadcast  $BPKT_i$  to any of their neighbors.

### 6.1 Handling of network failure in the TEMB protocol

Handling of network failure in the TEMB protocol is an easy task. The TEMB protocol will use the network failure mechanism that is incorporated within the underlying asynchronous duty-cycled MAC protocol. A forwarding node  $s$  will simply retransmit the  $BPKT_i$  after a retransmit timeout in case of a lost  $BPKT_i$  or  $ACK_i$ . Similar to BPEMB, if the receiver node does not respond within the maximum number of allowed retransmission attempt, the action depends upon the underlying duty-cycled MAC protocol.

## 7 Simulation results

We used a discrete event simulator Omnet++ [49], to evaluate the performance of the BPEMB and TEMB protocols. We compare the broadcast performance of BPEMB and TEMB protocols with EMBA and B-MAC protocols. The performance of the BPEMB, TEMB and EMBA protocols are evaluated using the preamble-based B-MAC protocol. In our simulations, we used networks of 100, 150, 200, 250, 300 and 350 nodes within a 1000 m × 1000 m area to check the effect of different network densities on the broadcast performance. For each network size, 15

topologies are randomly generated, which means that there are 15 runs for each network size. The parameters used in the simulation are summarized in Table 2. All the results mentioned in this section for each network size are the averages of 15 runs. The reason for varying the network density is to investigate the effects of sparse and dense networks on the performance of our proposed broadcast protocols. The *hello procedure* is executed by each node every 15 s. Within each network size of  $n$  nodes,  $n$  broadcast packets are broadcasted by each node at a random time within the simulation time.

### 7.1 Results of the performance comparison

The metric broadcast packet cost ratio (*BPCR*) is defined as the ratio between the actual number of transmissions required to propagate  $m$  broadcast packets and the minimum number of transmissions to propagate  $m$  broadcast packets. The *BPCR* is calculated as:

$$BPCR = \frac{t}{m(n-1)} \quad (5)$$

where  $t$  is the actual number of transmissions required to propagate  $m$  broadcast packets to each node within the network. The denominator value  $m(n-1)$ , is the total number of minimum unicast transmissions required to propagate  $m$  broadcast packets to  $n$  nodes in an error-free network. The *BPCR* metric indicates how many transmissions are used to propagate a broadcast packet to each node in a network. If the *BPCR* value is 1, a network has received the broadcast packet with the minimum number of transmissions. To measure the *BPCR*, we do not include any control packets (preamble packets and acknowledgements packets) including *hello messages*.

Figure 4 shows the performance of the BPEMB, TEMB, EMBA and B-MAC protocols in terms of *BPCR*. As the network density increases, the *BPCR* of each protocol decreases. The TEMB protocol achieves the best *BPCR* performance among all the protocols. As the network becomes denser, the *BPCR* difference between the BPEMB and TEMB protocols decreases. In denser networks, the

**Table 2** Simulation parameters

Parameter	Value
Simulation time	100 s
Network size	100, 150, 200, 250, 300, 350 (nodes)
Network area	1000 m × 1000 m
Transmission range	100 m
Slot duration	03 s
Maximum transmission attempts	6
Size of beacon	10 B
Size of ACK	10 B

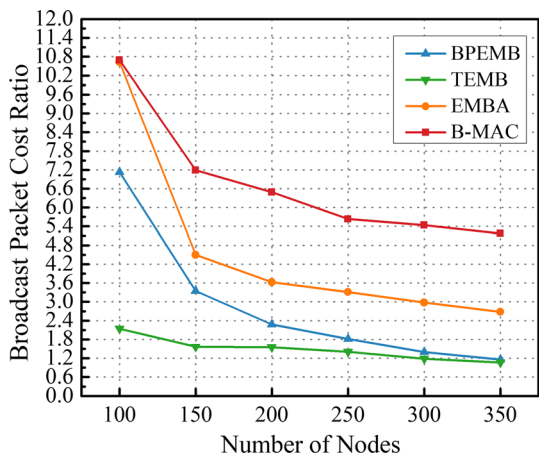


Fig. 4 The broadcast cost ratio (BPCR)

BPCR value of BPEMB and TEMB protocols starts approaching 1. Overall, the BPEMB protocol improves BPCR metric by 33.2–77.6% and 25.6–56.7% over the B-MAC and EMBA protocols, respectively. The TEMB protocol improves the BPCR value by 75–79.9% and 57.3–79.8% over the B-MAC and EMBA protocols, respectively. Effectively controlling the redundant messages has a very large impact on the BPCR value, as shown in the BPCR performance comparison. In the case of the TEMB protocol, the increase in the network density has no major impact on the BPCR value. The B-MAC and EMBA protocols are both effective in controlling the redundant messages, which led to the major improvement in the BPCR values over the B-MAC and EMBA protocols, as shown in Fig. 4.

Figure 5 shows the average number of redundant transmissions for each of the compared protocols. The TEMB protocol has sent the minimum number of redundant transmissions among the BPEMB, EMBA and B-MAC protocols. The BPEMB and TEMB protocols sent 65.2–93.5% and 94.5–97.4% less redundant packets than the B-MAC protocol, respectively. Compared with EMBA protocol, the BPEMB and TEMB protocols sent 58.8–83.5% and 89–96.9% less redundant packets than the EMBA protocol, respectively. The number of redundant packet transmission using the BPEMB protocol does not increase as sharply with increases in network density compared with the TEMB, B-MAC and EMBA protocols. The B-MAC protocol has a larger number of redundant transmissions because it does not consider which nodes have already received or will eventually receive the broadcast packet from one of their respective neighbors. The EMBA protocol is able to avoid a large number of redundant packets due to the forwarder’s guidance mechanism that it employs to take advantage of the existing quadrangular topologies in the network. The reduction in redundant packet transmissions helps to reduce the energy

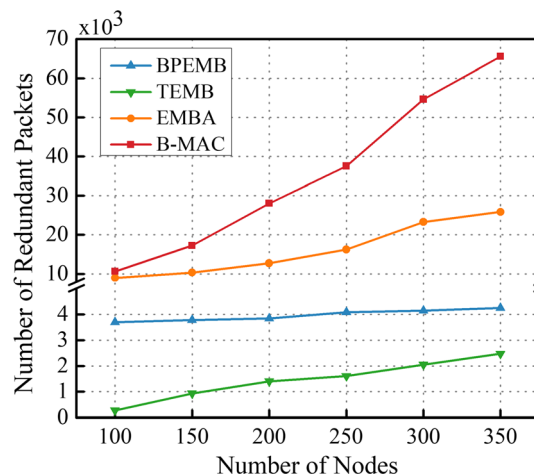


Fig. 5 The average number of redundant transmissions

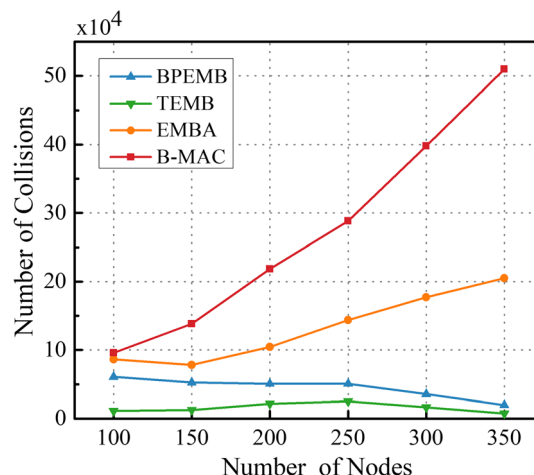


Fig. 6 The average number of collisions

consumption due to the reduced idle listening time. A lower number of redundant transmissions also helps to keep the network load to a minimum level.

Figure 6 depicts the average number of collisions of the compared protocols. The B-MAC protocol has an excessive number of collisions compared to the other protocols. In the B-MAC protocol, a receiver node simultaneously receives the broadcast packet from its neighboring nodes who have a copy of the broadcast packet, which leads to a collision at the receiver node. However, this scenario is prevented in the BPEMB protocol through the knowledge of the broadcast progress. In the TEMB, protocol, only the neighboring node that has the best link quality to the receiver node transmits the broadcast packet to it.

Figure 7 shows the average number of bytes transmitted in a network for each of the compared protocols, the BPEMB, TEMB, EMBA and B-MAC protocols. The average number of bytes transmitted includes all types of packets, such as the preamble packets, acknowledgment

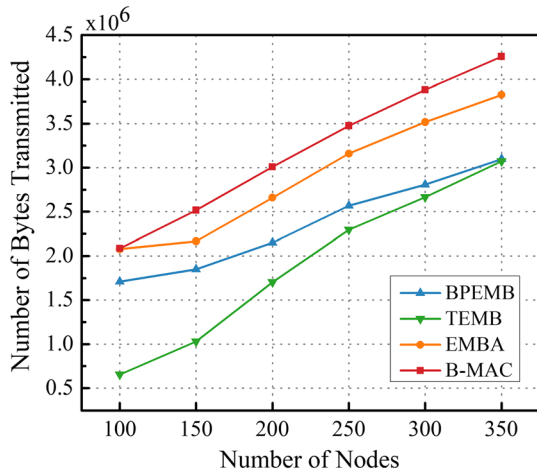


Fig. 7 The average number of bytes transmitted

packets and the hello messages exchanged during the simulation time. For all the network sizes, the BPEMB and TEMB protocols finish the broadcast procedure with a network load lower than that of both the EMBA and B-MAC protocols. When the BPEMB protocol is employed as a multihop broadcast protocol, the average number of bytes transmitted are 17.9–27.2% and 14.6–20.2% less than the B-MAC and EMBA protocols, respectively. In the case of the TEMB protocol, the average number of bytes transmitted are 27.7–68.5% and 19.6–68.4% less than the B-MAC and EMBA protocols, respectively. The B-MAC and EMBA protocols both have more redundant transmissions than the BPEMB and TEMB protocols, which contributed to the B-MAC and EMBA protocols transmitting more bytes.

Figure 8 shows the broadcast latency of all the compared protocols. Broadcast latency is defined as the time from the start of the broadcast packet from the source node to the time each node in the network has received the broadcast packet. The broadcast latency of the EMBA protocol increases as the network becomes denser. The duty cycle is the major factor in the broadcast delay in asynchronous duty-cycled MAC protocols. As the broadcast packet propagates from hop to hop in the asynchronous environment, a sender node must wait for the receiver node to wake up for a successful transmission. Another factor contributing to the delay is the retransmission of the broadcast packet due to collision at the receiver node. The BPEMB and TEMB protocols have 10.1–28.6% and 16.3–73.7% less broadcast latency than the B-MAC protocol, respectively. When compared with the EMBA protocol, the BPEMB and TEMB protocols achieves 7.7–25.2% and 19.3–72.4% less broadcast delay, respectively.

Figure 9 presents the average duty cycle performance of the BPEMB, TEMB, EMBA and B-MAC protocols. The

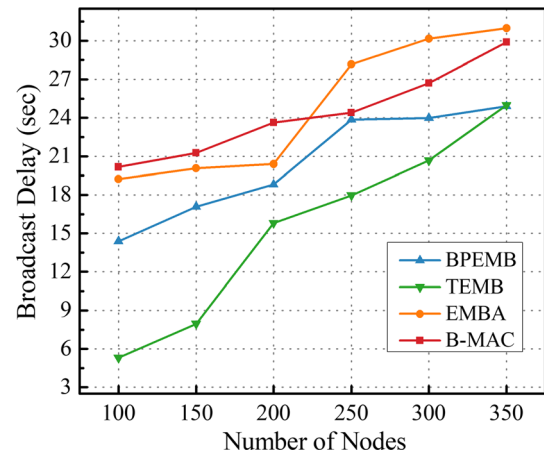


Fig. 8 The average broadcast delay

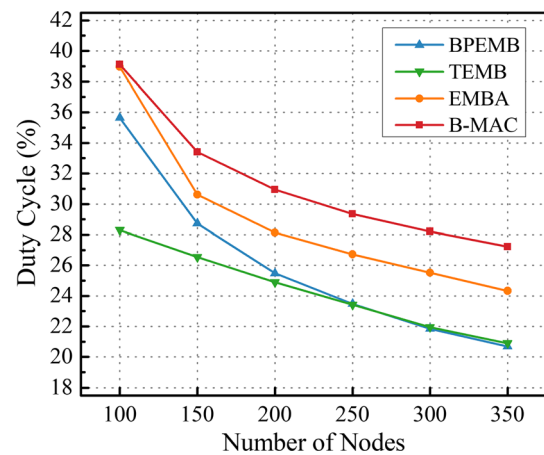


Fig. 9 The average duty cycle

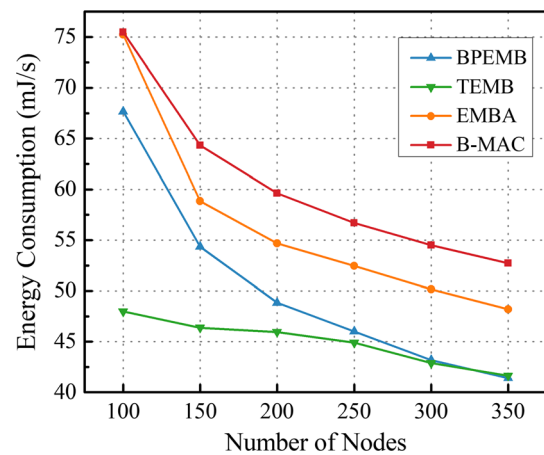


Fig. 10 The average energy consumption per node per second

average duty cycle for each of the protocols decreases as the network density increases. The BPEMB protocol achieves an 8.9–24% improvement compared with the B-MAC protocol and 6.1–14.9% improvement compared

with the EMBA protocol. On the other hand, the TEMB protocol achieves a 19.5–27.6% improvement when compared with B-MAC protocol and 11.5–27.3% compared with the EMBA protocol.

Figure 10 shows the average energy consumed by each node per second for the BPEMB, TEMB, EMBA and B-MAC protocols. The TEMB protocol consumes the least amount of energy in the sparse network, but as the network becomes denser, the BPEMB protocol consumes the least amount of energy. Overall, the energy consumed per node per second decreases as the network size increases. The energy consumption is closely related to the duty cycle of the nodes, as the duty cycle decreases, so does the energy consumption. The average energy consumption per node per second of the BPEMB protocol is 11.5–27.3% less than that of B-MAC protocol and 7.6–14% less than that of EMBA protocol. For the TEMB protocol, the average energy consumption per node per second is 20.7–36.4% less than that of B-MAC protocol and 13.5–36.2% less than EMBA protocol. One of the dominant factors that influences the energy consumption in multihop broadcasts is redundant transmission. The B-MAC and EMBA protocols have more redundant transmissions, and thus consume more energy than the BPEMB and TEMB protocols.

## 8 Conclusions

In this study, two energy efficient multihop broadcast routing protocols for asynchronous duty-cycled wireless sensor networks, the BPEMB and TEMB protocols, are proposed. Both of the protocols consider the link quality information to send broadcast messages over links with better quality. This approach avoids retransmission due to lost messages, which causes increased energy usage. In the BPEMB protocol, piggybacking the broadcast packet with partial information about the broadcast progress leads to better performance by reducing the number of redundant broadcast packets and decreasing the duty cycle. The simulation results show that both the BPEMB and TEMB protocols outperform the EMBA and B-MAC protocols in terms of broadcast packet cost and network load. Both the BPEMB and TEMB protocols decrease the redundant transmission of broadcast packets and reduce the duty cycles of the nodes, thereby leading to lower energy consumption. In our future work, we will take the load balance and mobility of nodes into consideration along with the link quality to reduce the broadcast latency and increase the energy efficiency.

## References

- Deif, D. S., & Gadallah, Y. (2014). Classification of wireless sensor networks deployment techniques. *IEEE Communications Surveys and Tutorials*, 16(2), 834–855. <https://doi.org/10.1109/Surv.2013.091213.00018>.
- Zebbane, B., Chenait, M., & Badache, N. (2017). A distributed lightweight redundancy aware topology control protocol for wireless sensor networks. *Wireless Networks*, 23(6), 1779–1792. <https://doi.org/10.1007/s11276-016-1248-5>.
- Zhai, C., Liu, J., Zheng, L., Xu, H. J., & Chen, H. (2012). Maximise lifetime of wireless sensor networks via a distributed cooperative routing algorithm. *Transactions on Emerging Telecommunications Technologies*, 23(5), 414–428. <https://doi.org/10.1002/ett.2498>.
- Habibi, J., Ghayeb, A., & Aghdam, A. G. (2013). Energy-efficient cooperative routing in wireless sensor networks: A mixed-integer optimization framework and explicit solution. *IEEE Transactions on Communications*, 61(8), 3424–3437. <https://doi.org/10.1109/Tcomm.2013.070213.120570>.
- Lu, Y., Comsa, I. S., Kuonen, P., & Hirsbrunner, B. (2016). Adaptive data aggregation with probabilistic routing in wireless sensor networks. *Wireless Networks*, 22(8), 2485–2499. <https://doi.org/10.1007/s11276-015-1108-8>.
- Jang, I., Yang, S., Yoon, H., & Kim, D. (2013). EMBA: An efficient multihop broadcast protocol for asynchronous duty-cycled wireless sensor networks. *IEEE Transactions on Wireless Communications*, 12(4), 1640–1650. <https://doi.org/10.1109/Twc.2013.022013.120477>.
- Wang, X. G., Wu, X., Zhang, X. M., & Liang, Y. C. (2018). An energy-efficient network-wide broadcast protocol for asynchronous wireless sensor networks. *IEEE Wireless Communications Letters*, 7(6), 918–921. <https://doi.org/10.1109/Lwc.2018.2840142>.
- Rajendran, V., Obraczka, K., & Garcia-Luna-Aceves, J. (2006). Energy-efficient, collision-free medium access control for wireless sensor networks. *Wireless Networks*, 12(1), 63–78. <https://doi.org/10.1007/s11276-006-6151-z>.
- Zhang, X. M., Zhang, Y., Yan, F., & Vasilakos, A. V. (2015). Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 14(4), 742–754. <https://doi.org/10.1109/Tmc.2014.2331966>.
- Mahapatra, C., Sheng, Z. G., Kamalinejad, P., Leung, V. C. M., & Mirabbasi, S. (2017). Optimal power control in green wireless sensor networks with wireless energy harvesting, wake-up radio and transmission control. *IEEE Access*, 5, 501–518. <https://doi.org/10.1109/Access.2016.2644607>.
- Liu, X. Y., Zhu, Y. M., Kong, L. H., Liu, C., Gu, Y., Vasilakos, A. V., et al. (2015). CDC: Compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(8), 2188–2197. <https://doi.org/10.1109/Tpds.2014.2345257>.
- Xiang, L., Luo, J., & Vasilakos, A. (2011). Compressed data aggregation for energy efficient wireless sensor networks. In *Proceedings of the 8th annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks* (pp. 46–54), IEEE. <https://doi.org/10.1109/SAHCN.2011.5984932>.
- Xu, X., Ansari, R., Khokhar, A., & Vasilakos, A. V. (2015). Hierarchical data aggregation using compressive sensing (HDACS) in WSNs. *ACM Transactions on Sensor Networks*, 11(3), 125–135. <https://doi.org/10.1145/2700264>.
- Basagni, S., Carosi, A., Melachrinoudis, E., Petrioli, C., & Wang, Z. M. (2008). Controlled sink mobility for prolonging wireless

- sensor networks lifetime. *Wireless Networks*, 14(6), 831–858. <https://doi.org/10.1007/s11276-007-0017-x>.
15. Tasharian, F., Moghaddam, M. H. Y., Sohraby, K., & Effati, S. (2015). On maximizing the lifetime of wireless sensor networks in event-driven applications with mobile sinks. *IEEE Transactions on Vehicular Technology*, 64(7), 3177–3189. <https://doi.org/10.1109/Tvt.2014.2354338>.
  16. Abo-Zahhad, M., Ahmed, S. M., Sabor, N., & Sasaki, S. (2015). Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks. *IEEE Sensors Journal*, 15(8), 4576–4586. <https://doi.org/10.1109/Jsen.2015.2424296>.
  17. Jakobsen, M. K., Madsen, J., & Hansen, M. R. (2010). DEHAR: A distributed energy harvesting aware routing algorithm for ad-hoc multi-hop wireless sensor networks. In *2010 IEEE international Symposium on "a world of wireless, mobile and multimedia networks" (WoWMoM)* (pp. 1–9), IEEE. <https://doi.org/10.1109/wowmom.2010.5534899>.
  18. Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), 537–568. <https://doi.org/10.1016/j.adhoc.2008.06.003>.
  19. Rout, R. R., & Ghosh, S. K. (2013). Enhancement of lifetime using duty cycle and network coding in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 12(2), 656–667. <https://doi.org/10.1109/TWC.2012.111412.112124>.
  20. Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(3), 493–506. <https://doi.org/10.1109/tnet.2004.828953>.
  21. Du, S., Saha, A. K., & Johnson, D. B. (2007). RMAC: A routing-enhanced duty-cycle MAC protocol for wireless sensor networks. In *IEEE INFOCOM 2007—26th IEEE international conference on computer communications* (pp. 1478–1486), IEEE. <https://doi.org/10.1109/INFCOM.2007.174>.
  22. Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on embedded networked sensor systems* (pp. 307–320), ACM. <https://doi.org/10.1145/1182807.1182838>.
  23. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 95–107), ACM. <https://doi.org/10.1145/1031495.1031508>.
  24. Huang, P., Xiao, L., Soltani, S., Mutka, M. W., & Xi, N. (2013). The evolution of MAC protocols in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(1), 101–120. <https://doi.org/10.1109/SURV.2012.040412.00105>.
  25. Sun, Y., Gurewitz, O., & Johnson, D. B. (2008). RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on embedded network sensor systems* (pp. 1–14), ACM. <https://doi.org/10.1145/1460412.1460414>.
  26. Sun, Y., Gurewitz, O., Du, S., Tang, L., & Johnson, D. B. (2009). ADB: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks. In *Proceedings of the 7th ACM conference on embedded networked sensor systems* (pp. 43–56), ACM. <https://doi.org/10.1145/1644038.1644044>.
  27. Zhang, X., Jia, X., & Yan, F. (2016). Dynamic delegation-based efficient broadcast protocol for asynchronous wireless sensor networks. *IEEE Communications Letters*, 20(6), 1195–1198. <https://doi.org/10.1109/LCOMM.2016.2550448>.
  28. Zhang, X., Yan, F., Li, C., & Ding, Q. (2016). Coverage efficiency-based broadcast protocol for asynchronous wireless sensor networks. *IEEE Wireless Communications Letters*, 5(1), 76–79. <https://doi.org/10.1109/LWC.2015.2498173>.
  29. Cheng, L., Niu, J. W., Luo, C. W., Shu, L., Kong, L. H., Zhao, Z. W., et al. (2018). Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks. *Computer Networks*, 134, 66–77. <https://doi.org/10.1016/j.comnet.2018.01.012>.
  30. Xu, L., Yang, G., Wang, L., Xu, J., & Wang, B. (2018). Opportunistic broadcasting for low-power sensor networks with adaptive performance requirements. *Wireless Networks*, 24(6), 2297–2317. <https://doi.org/10.1007/s11276-017-1473-6>.
  31. Sinha, A., Tassiulas, L., & Modiano, E. (2019). Throughput-optimal broadcast in wireless networks with dynamic topology. *IEEE Transactions on Mobile Computing*, 18(5), 1203–1216. <https://doi.org/10.1109/TMC.2018.2846246>.
  32. Duc, T. L., Le, D. T., Zalyubovskiy, V. V., Kim, D. S., & Choo, H. (2017). Towards broadcast redundancy minimization in duty-cycled wireless sensor networks. *International Journal of Communication Systems*, 30(6), e3108. <https://doi.org/10.1002/dac.3108>.
  33. Yan, F., Zhang, X., & Zhang, H. (2017). Efficient multihop broadcasting with network coding in duty-cycled wireless sensor networks (NET). *IEEE Sensors Letters*, 1(5), 1–4. <https://doi.org/10.1109/LESENS.2017.2756065>.
  34. Le, D. T., Duc, T. L., Zalyubovskiy, V. V., Kim, D. S., & Choo, H. (2017). Collision-tolerant broadcast scheduling in duty-cycled wireless sensor networks. *Journal of Parallel and Distributed Computing*, 100, 42–56. <https://doi.org/10.1016/j.jpdc.2016.10.006>.
  35. Liu, X., Li, G., Zhang, S., & Liu, A. (2018). Big program code dissemination scheme for emergency software-define wireless sensor networks. *Peer-to-Peer Networking and Applications*, 11(5), 1038–1059. <https://doi.org/10.1007/s12083-017-0565-5>.
  36. Xu, L. J., Yang, G., Xu, J., Wang, L., & Dai, H. P. (2018). Achieving adaptive broadcasting performance tradeoff for energy-critical sensor networks: A bottom-up approach. *Computer Networks*, 136, 155–170. <https://doi.org/10.1016/j.comnet.2018.03.007>.
  37. Cao, Z. C., Liu, D. B., Wang, J. L., & Zheng, X. L. (2017). Chase: Taming concurrent broadcast for flooding in asynchronous duty cycle networks. *IEEE/ACM Transactions on Networking*, 25(5), 2872–2885. <https://doi.org/10.1109/Tnet.2017.2712671>.
  38. Yan, F., Zhang, X., & Zhang, H. (2018). Flooding with network coding under a schedule-based spanning tree in low-duty-cycle wireless sensor networks. *IEEE Wireless Communications Letters*, 7(2), 270–273. <https://doi.org/10.1109/LWC.2017.2771357>.
  39. Belghith, A., Mosbah, M., Jemili, I., & Ghrab, D. (2017). Context-aware broadcast in duty-cycled wireless sensor networks. *International Journal on Semantic Web and Information Systems*, 13(3), 48–67. <https://doi.org/10.4018/ijswis.2017070103>.
  40. Yu, S., Liu, X., Liu, A., Xiong, N., Cai, Z., & Wang, T. (2018). An adaption broadcast radius-based code dissemination scheme for low energy wireless sensor networks. *Sensors*, 18(5), 1509. <https://doi.org/10.3390/s18051509>.
  41. Zhang, R., Mounsla, H., Yu, J., Chen, L., & Mehaoua, A. (2017). Multichannel broadcast in duty-cycling WBANs via channel hopping. *IEEE Internet of Things Journal*, 4(6), 2351–2361. <https://doi.org/10.1109/JIOT.2017.2767018>.
  42. Lu, D., & Dong, S. (2017). A neighbor knowledge and velocity-based broadcast scheme for wireless ad hoc networks. *International Journal of Distributed Sensor Networks*, 13(11), 1550147717743699. <https://doi.org/10.1177/1550147717743699>.
  43. Klues, K., Hackmann, G., Chipara, O., & Lu, C. (2007). A component-based architecture for power-efficient media access control in wireless sensor networks. In *Proceedings of the 5th*

*international conference on embedded networked sensor systems* (pp. 59–72), ACM. <https://doi.org/10.1145/1322263.1322270>.

44. Zhang, X. M., Wang, E. B., Xia, J. J., & Sung, D. K. (2013). A neighbor coverage-based probabilistic rebroadcast for reducing routing overhead in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 12(3), 424–433. <https://doi.org/10.1109/TMC.2011.277>.
45. Couto, D. S. J. D., Aguayo, D., Bicket, J., & Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4), 419–434. <https://doi.org/10.1007/s11276-005-1766-z>.
46. Fonseca, R., Gnawali, O., Jamieson, K., & Levis, P. (2007). Four-bit wireless link estimation. In *Proceedings of the 6th workshop on hot topics in networks*, ACM SIGCOMM.
47. Baccour, N., Koubâa, A., Youssef, H., Ben Jamâa, M., do Rosário, D., Alves, M., et al. (2010). F-LQE: A fuzzy link quality estimator for wireless sensor networks. In *Proceedings of the 7th European conference on wireless sensor networks* (pp. 240–255), Springer. [https://doi.org/10.1007/978-3-642-11917-0\\_16](https://doi.org/10.1007/978-3-642-11917-0_16).
48. Calinescu, G. (2003). Computing 2-hop neighborhoods in ad hoc wireless networks. In *International conference on ad-hoc networks and wireless* (pp. 175–186), Springer. [https://doi.org/10.1007/978-3-540-39611-6\\_16](https://doi.org/10.1007/978-3-540-39611-6_16).
49. OMNeT++ Discrete Event Simulator. (2018). <https://www.omnetpp.org/>.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Parmod Singh** received his Ph.D. degree from National Chiao Tung University, Hsinchu, Taiwan in 2019. He received his Master of Computer Applications degree from Guru Gobind Singh Indraprastha University, Delhi and B.Sc. degree in mathematics from University of Delhi. His research interests include Wireless Sensor Networks, Internet Protocols, and Ad Hoc Networks.



**Yaw-Chung Chen** received his B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, his M.S. degree from Texas A& M University, Kingsville, Texas, USA, and his Ph.D. degree from Northwestern University, Evanston, Illinois, USA. In 1986 he joined AT & T Bell Laboratories, where he worked on various exploratory projects. He joined National Chiao Tung University, Hsinchu, Taiwan as an associate professor in 1990.

He is currently a professor in the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. He is also the director of Information Industry Institute/National Chiao Tung University Joint Research Center. His research interests include Internet Protocols, fast mobility management, P2P systems and green computing. He is a senior member of IEEE and a member of ACM.